

## LOGICAL AND CONSTRAINT BASED BROWSE HIERARCHY WITH PROPAGATION FEATURES

Scott Bonneau  
Michael Nonemacher  
Jeremy Weinrib

5

### ABSTRACT OF THE DISCLOSURE

A logical and constraint-based hierarchical approach provides a highly flexible and expressive way in which to browse items stored in a database. The logical and constraint-based approach permits a user to create an arbitrary number of hierarchical representations of the items in a database. The approach permits items to be logically grouped on one level and grouped based on attribute/value constraints on the next. The hierarchical representation consists of nodes that are related to one another in a tree-like structure starting with a root node. Each node has a unique label, preferably indicative of the items in the database that it represents. Each node has a list of the labels of the nodes that are its children. Each node can only be the child of one parent node. Nodes may optionally express constraints based on attributes and their values that serve to define the scope of database items that fall under the node in the hierarchy. Each node inherits the constraints of its ancestors and therefore the scope of items in a database that that fall underneath a given node are defined by the aggregation of the constraints from the given node up to and including the root. A node that does not specify constraints instead specifies a logical grouping of items that cannot be specified by one or more constraints. The constraints are implicitly ANDed together in the aggregate. A logical grouping represents the equivalent of a logical ORing of constraints, but without the need for specifying attribute values or performing the function. A user browsing a database using the hierarchy triggers a search of the database by selecting one of the leaf nodes in the hierarchy. The constraints are aggregated and a rule is established that includes all of the database items that meet the aggregation of constraints. A database query is derived from the rule, and the database is searched for all items that meet the aggregated constraints. A set of items is returned and displayed for the user.

30